LArSoft continuous integration system requirements

Mark Dykstra, Patrick Gartung, Lynn Garren, Marc Mengel, Gianluca Petrillo, Erica Snider

Fermilab

Draft version 0.5

July 8, 2014

1.1 Abstract

This document will outline definitions and requirements for a “Continuous Integration” build and validation system for LArSoft and related experiments, which will automatically build and/or validate the LArSoft and/or experiment software when new updates are pushed into the central version control system for related software pacakges, using the Central Build Service being provided by the Fermilab Computing Division.

1.2 Assumptions

The requirements and specifications listed below are built on the assumption that the available continuous integration (CI) system will conform to the Central Build Service system requirements as documented in CD-DocDB-5319.

1.3 Definitions

These are terms we will be using later in the document:

Build Step: the smallest increment of execution provided by the CI system

Project: a set of build steps that the CI system runs as a single, self-contained unit.

Trigger: an event that initiates execution of a project.

Integration task: the basic unit of CI testing available to end-users.

Workflow: a set of steps executed as part of an integration task. The steps may be part of a single project, or a set of projects that are run in a sequence specified by triggers defined internally or externally to the projects

CI job: the execution phase of a project.

CI application: the software application that automates CI workflow triggering and execution.

1.4 CI System Requirements

1.4.1 Triggering

1.4.1.1 Trigger Input

It must be possible to specify input data and configuration data sources used in a project in the triggering event.

Projects may also specify input and configuration data sources within the project definition that are independent of any trigger parameters

1.4.1.2 Parameters

(*Desirable*) A project should be able to define parameter values at run time that can then be used in triggers called within the project.

1.4.1.3 Validation

The system should validate input parameters prior to attempting to queue or run the build/validation job.

1.4.1.4 Delay

There should be a configurable delay between the time of a trigger event and the launch of the triggered workflow.

1.4.1.5 Actions

There must be able to be triggers on at least the following actions:

a. push to develop branch on central repository, which will initiate a build and test of code on the head of the develop branch
b. push to master branch on central repository, which will initiate a build and test of code on the head of the master branch
c. successful completion of nightly build
d. successful completion of integration release build (if automated with the CI system)
e. successful completion of production release build (if automated with the CI system)
f. arbitrary manual triggers

1.4.2 Logging

1.4.2.1 Archive input/config

It must be possible to archive input and configuration data in a manner that supports the same level of version control as that used by the software being tested.

a) There must be a tool or suitably easy method for understanding the mapping of this versioned data to the specific versions of code to which is applies, and visa versa.

1.4.2.2 Tags

The system should provide a unique, easily distinguishable tag that identifies all CI jobs that were run and results obtained within a given workflow.

1.4.3 Data Transfers

1.4.3.1 Inputs

Input data may be in the form of parameter values passed to the project with the trigger, or via some generally available (on all relevant platforms), grid-compliant transfer protocol.

1.4.3.2 Outputs

Output data must be transferred via generally available (on all relevant platforms) grid compliant tranfer protocols. If a given project triggers another project, output data may be exported tot the triggered project via parameter values passed with the trigger.

1.4.3.3 Credentials

The system must manage any credentials required to perform input and output data transfers.

1.4.3.4 Encryption

All data channels, including parameters passed via triggers, must be encrypted.

1.4.3.5 Manual

It must be possible for authorized users to manually trigger projects or workflows via command line or GUI

1.4.3.6 Propogation

All relevant configuration information in a trigger event must be propagated to subsequent triggers within a given workflow.

1.4.4 Workflows

1.4.4.1 Versionable

Workflows that build software must support custom, per-build specification the branch, tag or commit to use for each repository to be included in the build.

a. Allow specification of a single “global” branch or tag, which directs the system to use that branch or tag in every repository in which it is found.

b. Use the following scheme to resolve branches, tags or commits to use in a build:
    - b.i. If a branch, tag or commit has been specified for a given repository, build the specified code in that repository. Failure to find the specified branch, tag or commit should result in an error condition.
    - b.ii. If no branch, tag or commit has been specified for a given repository, use the global branch or tag if defined and present in the repository.
    - b.iii. If no global branch or tag has not been specified or is not found in the given repository, then default to the head of the develop branch.

c. All software to be built must be pulled from the central repository, or from some other secure source repository.

d. (*Desirable*) For LArSoft builds, it would be desirable to have a command that examines the "active" branches in a user's work area, then launches a build of those branches on the central repository.

1.4.5 Independence

1.4.5.1 Tests

All tests must be executable independently of the CI service.

1.4.5.2 Builds

All Builds must be executable independently of the CI service.

1.4.6 Display

1.4.6.1 Real-time

(Desirable) Real-time visualization of build-step progress and results within a project and results within a workflow is highly desirable.

1.4.7 Integration/Development

1.4.7.1 Migration

The system should allow for simple migration of workflows and projects from a development/integratino environment into production

a) No changes to the test scripts or CI application configuration should be needed in order to accomplish this migration.

b) Results from projects or workflows under development or integration testing should be visible from the production CI application.

1.4.8 Platforms

1.4.8.1 All Supported

All CI jobs and workflows must run on all supported platforms at any site that runs any of those platforms.

1.4.9 Validation Testing

1.4.9.1 Dynamic

The CI system must have a mechansim for identifying the proper tests to execute at run time.

a) It must be possible to add or remove tests without modifying the CI application configuration or any CI system scripts.

1.4.9.2 Format

Stand-alone tests must be in the form of an executable script located under a dedicated test directory in the appropriate software repository (version control system).

1.4.9.3 Suites

The CI system must support a mechanism for associating specified groups of tests that define a “test suite”. A single trigger event should result in the execution of a particular test suite.

1.4.9.4 Time Estimates

(Desirable) Tests should have an optional standardized execution time score that allows the system to estimate the CPU required to run the test on a given platform.

1.4.9.5 Timeouts

(Desirable) The system should provide a means to force test failure if the run time is:

a. below some user-defined, per-test minimum CPU time;
b. above some user-defined, per-test maximum CPU time threshold that is scaled to the time score;
c. above some user-defined, per-test maximum absolute CPU time threshold;
d. or above some system-defined, per platform absolute CPU time threshold

1.4.9.6 Memory Estimates

Tests should have an optional memory score allows the system to estimate the expected maximum memory size of the test during execution.

1.4.9.7 Memory limits

(*Desirable*) The system should provide a means to force a test to fail if the memory consumption is:

e. above some user-defined, per-test maximum memory size threshold that is scaled to the memory score
f. above some user-defined, per-test absolute memory size threshold
g. above some system-defined, per platform absolute memory size threshold

1.4.9.8 Parallelism

The CI system should allow parallel execution of tests when possible